Design and Optimization of a Conceptual Automated Yard using Overhead Grid Rail System

Elias B. Kosmatopoulos, Chin Liu and Petros Ioannou
Center for Advanced Transportation Technologies
EEB200B, MC2562
University of Southern California
Los Angeles, CA 90089

14 June 2002

METRANS
Transportation Center
USC / CSULB

Disclaimer

The Contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation Centers Program, and California Department of Transportation in the interest of information exchange. The U.S. Government and California Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California or Department of Transportation. This report does not constitute a standard, specification, or regulation.

Abstract

Booming world trade, scarcity of land for yard expansion in many ports, and deployment of new massive megaships have magnified the need for finding better ways of performing container terminal operations. High-density, automated container terminals are currently considered as a candidate to improve the performance of container terminals. The overhead Grid Rail (GR) system is such a candidate, offering the advantages of high storage density, fast loading/unloading, flexibility and reliability and no interference between manual and automated operations. Moreover, contrary to other automated container concepts, the simplicity of GR operations makes the development of optimal or nearly optimal dispatching algorithms possible.

The purpose of this study is to design and optimize an automated container yard that is equipped with a GR system (consisting of a number of GR units) for storage and retrieval of containers in the yard. Algorithms will be developed for the optimal choice of the GR system parameters (number of GR units and number of shuttles) as well as for optimal dispatching schedules.

Table of Contents

List of Figures

List of Tables

1. Introduction

Booming world trade, scarcity of land for yard expansion in many ports, and deployment of new massive megaships have magnified the need for finding better ways of performing container terminal operations. High-density, automated container terminals are currently considered as a candidate to improve the performance of container terminals. Different high-density automated container yard concepts based on the use of Automated Guidance Vehicles (AGVs), linear-motor conveyance shuttles, automated multi-story retrieval and storage units and overhead Grid RAIL (GR) units have been proposed as possible candidates for future automated container terminals. The overhead Grid RAIL (GR) offers the advantages of high storage density, fast loading/unloading, flexibility and reliability and no interference between manual and automated operations. Moreover, contrary to other automated container concepts, the simplicity of GR operations makes it possible to develop optimal or nearly optimal dispatching algorithms.

Automated container yard operations possess complicated dynamics that makes it very difficult to construct algorithms for optimizing the system parameters (e.g., number of shuttles) and the dispatching schedules. Most of the techniques in the literature are based on heuristics and on simulations of today's terminal practice (see [3]-[9]). Optimization techniques such as dynamic programming, simulated annealing and genetic algorithms have been applied [1]-[10], [12]-[14]. However, these techniques are known to suffer from slow convergence, convergence to local minima, heavy computations and sensitivity to the initial guess. Mixing heuristics with optimization techniques often leads to solutions that are better than the ones based on heuristics alone.

A simulation model was developed by August Design for a simple GR system [11]. In this simulation model, the dispatching algorithms are based on an expert system. This expert system has initially been developed for emulating today's practice. It was fine tuned afterwards by evaluating different strategies by means of simulations.

The goal of this study is to develop algorithms for optimizing the GR parameters and operations and to perform a productivity analysis of GR systems. The main findings/contributions of this study are summarized below:

- The concepts of GR unit and GR system are introduced. Based on the observation that smaller GR units have better performance characteristics than larger ones, we propose the concept of GR system consisting of many simple GR units. The introduction of this concept simplifies considerably the optimization algorithms and analysis of the GR operations.
- A new optimization algorithm for dispatching is introduced. The algorithm is computationally simple and efficient and, contrary to prior algorithms, guarantees solutions that are nearly optimal. Moreover, the proposed algorithm can be implemented real-time.

- Simulation studies are performed on a typical container yard configuration and it is shown that:
  1. The GR system productivity increases considerably as the number of GR units increases
  2. The GR system can achieve high productivity rates, outperforming 2 and 3 times the productivity of existing systems; more precisely, GR systems using the dispatching algorithm proposed are capable of achieving productivity rates as high as 300 containers per crane/per hour even when reshuffling is needed
  3. The proposed dispatching algorithm considerably improves the GR productivity, especially in the case where reshuffling is needed.

- One of the crucial problems in today's maritime transportation is the development of appropriate cargo handling technologies that will make possible the loading/unloading of a ship as fast as possible in many cases less than 24 hours. In our work, we evaluate the capabilities of GR systems to serve such a purpose when they are used in automated container yard terminals. Using simulation studies, we show that an automated container yard whose basic element is a GR system can achieve the aforementioned goal.

2. GR Unit

A GR unit, shown in Figure 1, proposed by Sea-Land and August-Design consists of an overhead rail, passive switches, shuttles, container buffers and a computer control system. Each of the major components is discussed below.

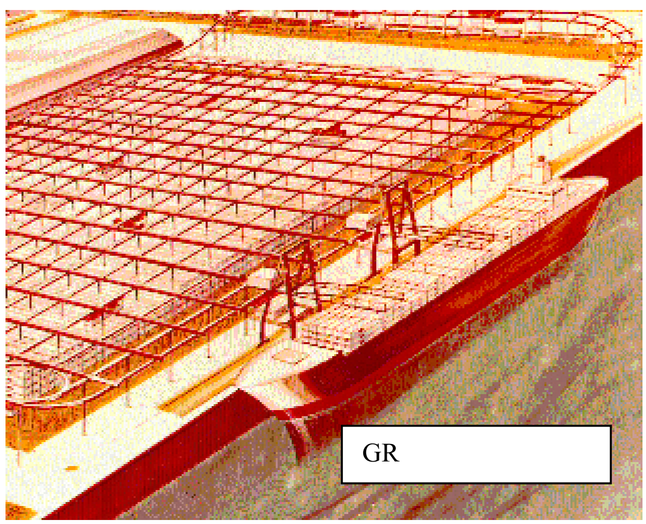

Figure 1: GR Configuration

The major components of a GR unit shown in Figure 1 include an overhead rail, passive switches, shuttles, container buffers and a computer control system. Each of the major components is discussed below.

- **Overhead Rail.**

The overhead grid rail provides service to the container yard, quay, rail spur and any other auxiliary storage areas. The rail also may extend to maintenance areas, the gates and the freight warehouse. The layout of the overhead rail is flexible and can be configured to ideally suit the needs of the terminal. The overhead rail also can be designed so that it can be removed and re-deployed at another terminal or at different areas of the same terminal (although such a task is not trivial).

- **Passive Switches.**

A major feature of GR is the method in which the shuttles can switch from one rail to another. Typically, overhead rail systems employ mechanical devices that physically move sections of the track in and out of the path of the shuttle. To continue in the current direction, a straight section is moved into place. To make a turn, a curved section of track is moved in front of the shuttle. Such switches are expensive, require a great deal of maintenance, and are slow, often requiring minutes for a single switching operation.

GR employs a unique passive switching scheme that requires no moving parts in the rail, and only a one-degree of freedom guide wheel on the shuttle. This permits the shuttle to steer itself throughout the grid rail system. The scheme requires that the shuttle's wheels pass over a gap in the rail as the shuttle goes through a switch. Moving over a gap is a challenging situation for a shuttle carrying a fully loaded container. However, design studies [11] indicate that it is feasible. Moreover, a physical scaled model built by August Design has proved the concept. The passive switch has many advantages: it is only slightly more expensive than a straight section of rail; it is lighter in weight; it requires low maintenance, and it permits rapid access to any section of the overhead rail.

- **Shuttles.**

The shuttles are the major container handling devices in the GR system. The shuttles can precisely move to any location covered by the GR unit in order to pick and place containers. Shuttles have access to different *container buffers* that are located in different places around the overhead rail. These container buffers (explained in more detail later on) are used to temporarily store containers. The shuttles pick/load the containers from/to the container buffers.

The shuttles are assigned a task by the computer control system. Such a task could be either to pick a container from the yard, move the container to a container buffer and load the container to the buffer, or vice versa, pick a container from a container buffer and load it at a specific position in the yard. Collision avoidance and traffic control of the shuttles are achieved by online monitoring the movement of the shuttles and by employing infrared techniques. Each shuttle shines infrared beacons as a warning to approaching shuttles and uses infrared receptors to sense the warning beacons of other shuttles. If a shuttle detects a warning beacon it will automatically slow down or stop to avoid a dangerous situation. When the beacon is cleared, the shuttle automatically resumes its task. The traffic is also monitored by the central computer control system. In the case where two shuttles meet at an intersection, the control system manages appropriately the movement of the shuttles in order to avoid a possible collision.

Different technologies can be used so the shuttles and the control system can determine the location of the shuttles in grid rail:

Bar Codes. One possibility – as the one used in the August Design scale model – is to use bar codes. In this case, each location of the yard is bar coded and the code is embedded in the overhead rail. Using the bar code scheme, the shuttles can repeatedly position themselves over any container in the system within approximately a tenth of an inch.
Transponders. This system uses transponders at the rail to transmit "labels" containing a unique ID and position data. Each time a vehicle encounters a line, it compares that information with its calculated position, then corrects itself. This navigation system is flexible and has accuracy up to ± 3 cm. It is disruptive to install, though, and transponders may shift in bad weather.
Laser Guidance. The vehicle uses laser beams and reflectors to calculate its distance from fixed points using triangulation. A minimum of three targets must be detected at each time during travel. Normally there should always be five visible targets. The vehicle can then use its onboard map to determine and correct its own location. This system is flexible, accurate, and imposes low infrastructure cost. However, it is affected by adverse weather, needs large number of reflectors, and has a long set-up time.
Millimeter Wave Radar (MMWR). A rotating MMWR detects the presence of beacons at known locations in the yard to determine the vehicle's position. The beacon observations are then processed to constantly update the vehicle's position. MMWR is accurate up to ± 10 cm, but it is expensive, needs long set-up time, and a large number of reflectors.
Dead Reckoning. In this method vehicle-mounted motion sensors precisely detect vehicle direction and speed. Given a known starting position, the integration of speed data over time allows the location of the vehicle to be determined. Because errors accumulate with distance traveled, these systems become inaccurate and unreliable unless the vehicle's position is periodically corrected by some other means. Dead Reckoning is simple, flexible, inexpensive, and easy to accomplish in real-time.

- **Container Buffers.**

The interface between the shuttles and the quay cranes, rail system, gate and other parts of the yard is the container buffer. Cranes are used to load/unload the container buffers with containers coming/going from/to other parts of the yard. The buffer accumulates containers and has the ability to move the containers into precise locations beneath the crane and the shuttle tracks. The buffers are high off the ground, at the same level as the overhead grid system, which has the effect of reduced hoist time of the shuttles.

In a typical GR unit there may be up to three different types of container buffers: quay buffers that serve the quay cranes, gate buffers that serve the gate, and rail buffers that serve the rail spur.

- **Computer Control System.**

The central computer control system communicates with the shuttles and the container buffers using a custom designed real time local area network. Different types of communication technologies can be used for the communication of the shuttles with the computer control system. The particular type of communication technology depends on the technology used for the positioning of the shuttles. For instance, in the case where bar codes are used an infrared communications medium can be used.

The computer control system is provided with a database with the location and the types of containers and a navigation algorithm for guiding the shuttles. The navigation algorithm uses the information provided by the shuttles, the container buffers and the database in order to assign the shuttles with tasks, control and monitor the movement of the shuttles and update the information stored in the database.

3. GR System

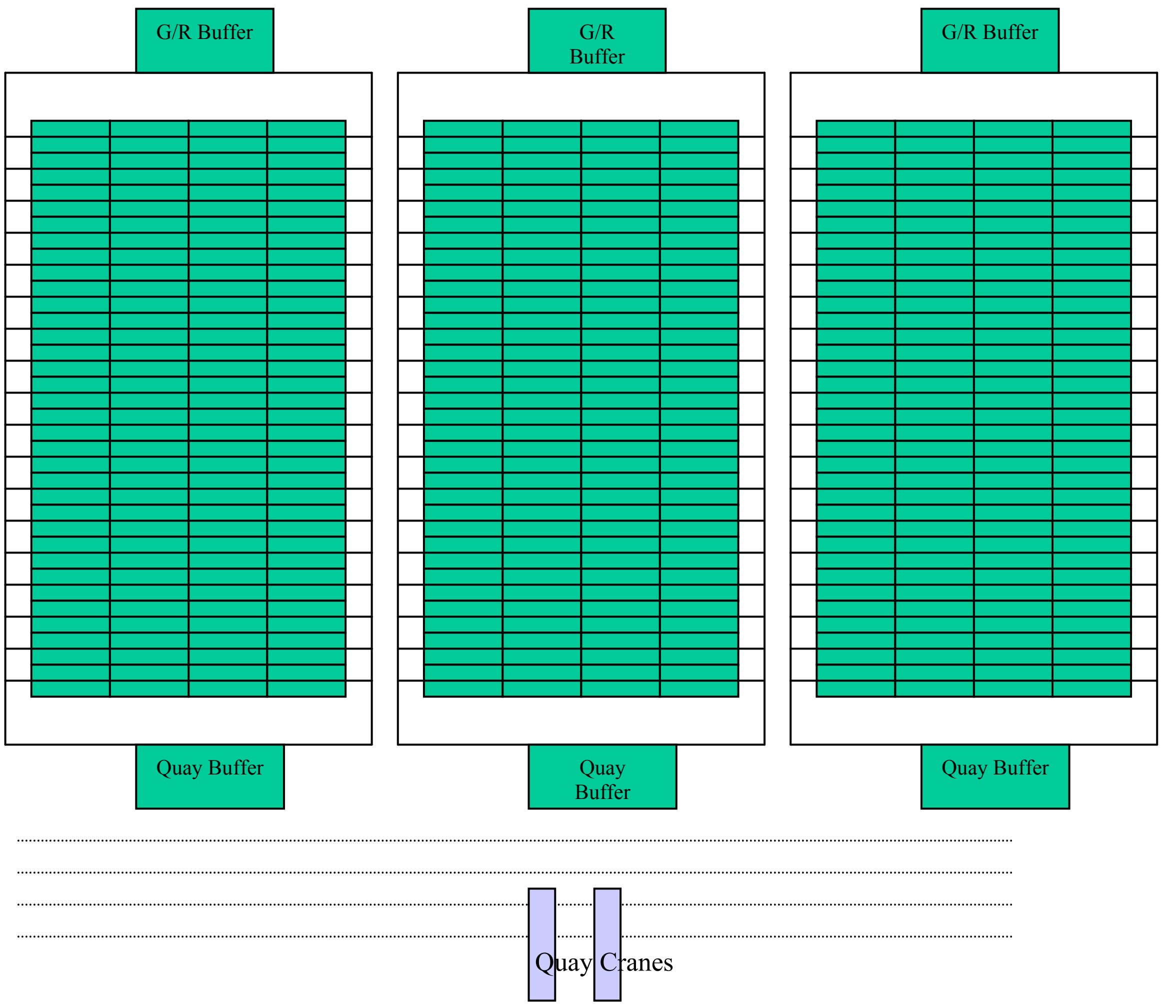

Figure 2: A GR System

A GR system consists of multiple GR units positioned parallel to each other as shown in Figure 2. The advantages of covering all or part of the yard with multiple GR units instead of a single GR unit are many:

- GR units that cover smaller areas have better performance characteristics than larger ones. In smaller GR units, the average distance covered by the shuttles is smaller, and thus the average shuttle travel time is smaller, too.
- The productivity of a GR unit heavily depends on the number of container buffers used. In general, if more than one buffer are used for feeding the same part of the yard (e.g., the quay cranes) then the following phenomenon occurs: one of the buffers is loaded/unloaded with the same rates as if in the case where only one buffer was used, while the other buffers are loaded/unloaded with very low rates. Therefore, it is preferable to use small GR units with one container buffer instead of big ones with more than one buffer.
- GR units that cover small areas can be deployed and removed more easily than large GR units.

In this work, we consider a GR system consisting of N GR units as shown in Figure 2. For simplicity and without loss of generality, we assumed that all N units have the same dimensions, and thus are capable of servicing the same number of containers. Moreover, we assume that each of the GR units is equipped with M shuttles. Each of the units is equipped with one *quay container buffer* located at the bottom of the unit and one *gate/rail container buffer (g/r buffer)* located at the top of the unit. The quay buffers are used to serve the quay cranes, the gate/rail buffers serve the gate and the rail.

The containers are loaded/unloaded by the shuttles at the buffers. Automated Guidance Vehicles (AGVs) load/unload the containers from the buffers and transfer them from/to the quay cranes, gate and the rail spur.

4. GR Operation

Below we describe the operations of each GR unit in the case where the containers are loaded from the yard to the ship. The cases where the containers are loaded from the ship to the yard, from the gate/rail to the yard, from the yard to the gate/rail are defined similarly.

Let us assume that there are K^j containers[1] in each unit to be loaded to the ship. Let $Y_i^j, i = 1,\dots,K^j, j = 1,\dots,N$ be a three-dimensional vector that denotes the position (the first two entries corresponds to the x-y coordinates of the container and the third entry denotes how high the container is placed in the stack) of the of the i-th container in the j-th GR unit. Then, the operations taking place in each GR unit are the following:

[1] In the sequel K^j will denote the number of containers left to be loaded.

1. A container in the yard is assigned to each shuttle by using an appropriate dispatching algorithm. The assignment of containers to shuttles depends on the current position of the shuttles and the position of the containers that are left to be loaded.
2. The shuttle moves to the position of the assigned container. Here it is assumed that the GR rails allow only one-directional motion, which is chosen arbitrarily to be counter-clockwise. Bi-directional motion is not considered because it will create more congestion and traffic problems and complicate navigation.
3. Collision avoidance between two shuttles is always avoided by having the shuttle with the larger index stop and wait until the other shuttle completes its move and eliminates the possibility of collision
4. Once the shuttle reaches the position of its assigned container it spends T_y seconds, which is the time needed for the shuttle to pick the container.
5. After the shuttle has picked the container, it moves towards the quay buffer. When it reaches the position of the quay buffer, it takes T_b seconds for the container to be loaded on the buffer.

The time-variables T_y and T_b depend on the shuttle capabilities to load/unload the buffer and to pick/load the container in the yard. Moreover, T_y depends on the numbers of containers that are above the assigned container – in the case where there are containers above the assigned container, reshuffling is needed by moving these containers to the neighboring stacks – as well as the distance (depth) between the shuttle and the assigned container. In general, T_y can be mathematically described as follows

$$T_y = F(Y_i^j, C_i^j)$$

where C_i^j is a three-dimensional vector whose first entry denotes the number of containers that are above the *i*-th container in the *j*-th GR unit and the other two entries denote the height of the container stack in the two neighboring stacks, and $F(\bullet)$ is a nonlinear function. The value of the function $F(\bullet)$ is obtained by simulating the shuttle operations of moving the hoist down to the container position, attaching the container to the hoist, moving the container and loading it to the next stack, return the shuttle back to the stack, etc, until the desired container is picked.

In our analysis, we assume that T_b is a known constant. Moreover, we assume that when the shuttle is picking a container, the hoist moves with known constant velocities $\overline{v}_{unloaded}, \overline{v}_{loaded}$ in the case where it is unloaded and loaded, respectively, and that it takes $\overline{T}_c$ seconds to attach the container to the shuttle's hoist, where $\overline{T}_c$ is a known constant. In the case where reshuffling is needed, we also assume that the shuttle is moving to the neighboring stack positions with known constant velocities $v_{unloaded}, v_{loaded}$ for the case where it is unloaded and loaded, respectively. It is worth noticing that the dispatching algorithm proposed in the next section is robust with respect to modeling errors or time-variations of the parameters $T_b, \overline{v}_{unloaded}, \overline{v}_{loaded}, \overline{T}_c, v_{unloaded}, v_{loaded}$. Moreover, in our simulations, the above parameters are random Gaussian variables with mean equal to the nominal value used in the dispatching algorithm development.

5. Optimization of GR Operations

Consider a GR system as the one described in the previous section. Also consider a given scenario for the operation of the AGVs used to load/unload the GR system by transferring containers to/from the quay ships, gate and rail spur[2]. Given this scenario, given a particular set of containers to be loaded/unloaded and assuming that the dimensions of the yard are fixed the total GR system throughput is a function of the numbers *N* of GR units, *M* of shuttles in each unit and the particular dispatching scenario used for handling the containers. By dispatching scenario, we mean the specific policy used for the assignment of containers to shuttles. In other words, given a dispatching scenario $S(Y^j)$ for handling the containers (here Y^j denotes the matrix whose columns correspond to the position vectors Y_i^j of all the container in the *j*-th GR unit that are to be loaded to the ship), the total throughput is given as

$$Throughput = Th(N, M, S(Y^1), \cdots, S(Y^N))$$

where *Th* is some nonlinear function.

We are interested in constructing an optimization algorithm for finding the integers *N, M* and the dispatching scenarios $S(Y^j)$ that maximize throughput. In general, the solution to such an optimization problem is difficult, to obtain. To make things worse, any optimizing algorithm must be computationally fast in order to generate solutions that can be implemented before the ship arrives. Existing algorithms for the solution of nonlinear optimization problems like the one stated above are known for their slow convergence or convergence to local minima and for their so-called "curse of dimensionality".

To overcome this problem we "break" the above optimization problem into two sub-problems:

Sub-problem 1: Assume a specific dispatching policy. Then, find the numbers N^*, M^* that maximize throughput under the aforementioned policy.

Sub-problem 2: Fix the numbers N^* of GR units and M^* of shuttles and find the dispatching policy that maximizes throughput.

There are many advantages of breaking the problem of optimizing throughput into the two above sub-problems. First, it is easier to solve two simpler problems instead of a complex one. Second, the solution of the second sub-problem is not required to be known as soon as the ship has arrived; If the first steps of the dispatching algorithm are known when the ship has arrived and the algorithm is fast enough to provide the appropriate dispatching commands on-time, then we can have the optimization algorithm running while the loading operations have started.

---

[2] Our analysis will be concentrated only in the case of loading containers from the yard to the quay buffer. The rest cases (loading from the quay buffer to the yard, loading from the gate/rail buffer to the yard, loading from the yard to the gate/rail buffer) can be treated similarly.

Solution to the Sub-Problem 1

Although a variety of optimization algorithms could be applied for the solution of this sub-problem, most suffer from slow-convergence or convergence to local minima. Moreover, most of the existing optimization algorithms require the explicit knowledge of the nonlinear function *Th*, which is very difficult to be obtained in our case. Instead of using an optimization algorithm, we can use **exhaustive search** methods by using simulations. In other words, we can simulate all the GR operations under a given dispatching scenario for all feasible values or combinations of N and M and find the values of N and M for which the simulations produced the maximum throughput. We notice here that since all GR units are identical, we don't need to simulate the operations of the whole GR system but we can rather simulate a single GR unit, saving thus a great amount of computational time.

Solution to the Sub-Problem 2

Once the number of GR units and shuttles are fixed, we can find the dispatching policy that maximizes throughput. A variety of optimization algorithms can be applied for the solution of this problem. These include dynamic programming, neural networks, simulated annealing, genetic algorithms and possibly others. However, these methods are computationally complex and they cannot, in general, guarantee convergence to optimal solution. Below we present a dispatching algorithm that is *nearly optimal* (The notion of "nearly optimal" will be made clear in the sequel).

Let us start with the following two observations:

Observation 1: Since one-directional motion is assumed the paths assigned to the shuttles by the dispatching algorithm are unique. That is, once a container is assigned to a shuttle there is only one path that the shuttle can follow to pick the container and bring it to the buffer.

Observation 2: The success of a dispatching policy depends on the total time the shuttles are waiting:
(D1) in the buffer queue,
(D2) in the intersections for the case where there is a possibility of collision and
(D3) inside the yard in the case where they cannot move because there is another shuttle in front of them picking a container.
A successful dispatching algorithm is one that keeps this total time as low as possible.

Note that in the case where there are no delays due to any of the factors (D1)-(D3) mentioned above, the total time T_k needed for a shuttle to travel to the container position (starting from the buffer position), pick the k-th container in the yard and bring it to the buffer is given as follows[3]

[3] In the sequel of this subsection, we will omit the GRAIL unit index j for simplicity.

$$T_k = \frac{d_{1k}}{v_{unloaded}} + F(Y_k, C_k) + \frac{d_{2k}}{v_{loaded}}$$

where $v_{unloaded}, v_{loaded}$ is the shuttle's speed when it is unloaded and loaded, respectively, and d_{1k}, d_{2k} are the total travel distances from the buffer to the container and from the container to the buffer, respectively. These two distances can be easily calculated for a given GR unit configuration.

Let us assume a dispatching algorithm that produces an assignment to all shuttles when there is an empty shuttle, which has just left the quay buffer area and no container is assigned to it (note that at this time, the other shuttles are moving loaded to the buffer or being unloaded at the buffer). Let t_{assign} denote the time-instant at which an assignment is taking place. Let also T_i denote the total time needed (starting at time t_{assign}) for the *i*-th shuttle to travel to the buffer and load the buffer assuming no delays due to factors (D1)-(D3), i.e.,

$$T_i = T_b + \frac{d_i}{v_{loaded}}$$

where d_i denotes the total travel distance from the *i*-th shuttle position at time t_{assign} to the buffer position. Then define the variable T_{ik} as follows

$$T_{ik} = T_i + T_k$$

The above variable denotes the time needed (starting at time t_{assign}) for the *i*-th shuttle to travel to the *k*-th container position, pick the container and bring it to the buffer in the case where there are no delays due to any of the factors (D1)-(D3).

Consider a dispatching algorithm that each time it produces an assignment the following conditions are satisfied (here the index k_ℓ corresponds to the container assigned to the shuttle i_ℓ):

(C1) $T_{i_1k_1} + (M-1)T_b \le T_{i_2k_2} + (M-2)T_b \le \cdots \le T_{i_{M-1}k_{M-1}} + T_b \le T_{i_Mk_M}, i_\ell \ne i_j, k_\ell \ne k_j \forall \ell \ne j$

(C2) In the case where $k_\ell, k_j, \ell \ne j$ correspond to two containers in the same rail row, then (assuming $\ell < j$) $\overline{T}_\ell < \widetilde{T}_j$ where $\overline{T}_\ell$ is the time needed for the shuttle i_ℓ to travel from its position at time t_{assign} to the position of container k_ℓ and pick the container (i.e., $\overline{T}_\ell$ equals $T_{i_\ell k_\ell}$ minus the time needed for the shuttle to reach the buffer starting at the container k_ℓ position) and $\widetilde{T}_j$ is the time needed for the shuttle i_j to travel from its position at time t_{assign} to the position of container k_ℓ.

Condition (C1) states that if container k_ℓ is assigned to the shuttle i_ℓ, then in the case where there are no delays due to factor (D3), there will be no possibility of two shuttles colliding in an intersection and the shuttles will spend no time waiting in the buffer queue. Condition (C2) makes sure that there are no delays due to the factor (D3). In other words, if a solution satisfying (C1), (C2) can be found then there will be no delays due to (D1)-(D3). Note also that in the case where the time T_y needed for a shuttle to pick, a container in the yard is constant for all containers (this is true when no reshuffling is needed and all the containers are at the same depth), and then condition (C2) is not needed.

Based on conditions (C1), (C2) we propose the following dispatching algorithm

Dispatching Algorithm (DA)

Step 1 Initialize all shuttle positions so that they are all at least T_b seconds[4] apart and they are moving loaded towards the buffer.

Step 2 Pick[5] randomly *M* containers that belong to different rows. If there are, no *M* containers that belong to different rows go to Step 3. Calculate the times T_i, T_k for the shuttles and the containers, respectively. Sort both T_i, T_k so they satisfy

$$T_{i_1} < T_{i_2} < \cdots < T_{i_M}$$
$$T_{k_1} \le T_{k_2} \le \cdots \le T_{k_M}$$

Assign container k_ℓ to shuttle i_ℓ.

Step 3 Pick randomly *M* containers. Calculate the times T_i, T_k for the shuttles and the containers, respectively. Sort both T_i, T_k so they satisfy

$$T_{i_1} < T_{i_2} < \cdots < T_{i_M}$$
$$T_{k_1} \le T_{k_2} \le \cdots \le T_{k_M}$$

Assign container k_ℓ to shuttle i_ℓ. Check if condition (C2) is satisfied. If no disregard the containers k_j that violate this condition, and repeat Step 3 by replacing the disregarded containers with new ones. If there[6] is no combination of shuttles/containers that satisfy (C2) assign *M* containers randomly to the shuttles.

We have the following result:

Proposition 1: The dispatching algorithm DA guarantees that there are no delays due to factors (D1)-(D3) as long as there are at least *M* containers in different rows.

Proof: Since there are at least *M* containers in different rows, the DA algorithm uses Step 2. Note also that since the shuttles have been initially at least T_b seconds apart, at the next assignment the following inequality is valid

$$T_{i_1} + (M-1)T_b \le T_{i_2} + (M-2)T_b \le \cdots \le T_{i_{M-1}} + T_b \le T_{i_M}$$

---

[4] We say that two shuttles are *T* seconds apart, when the second shuttle needs *T* seconds to reach the first shuttle position.

[5] In the case where the number of containers *K* left to be loaded is less than *M,* we add *M-K* dummy container in the yard and assign *M-K* shuttles to these fictitious containers.

[6] Instead of assigning containers randomly to the shuttles, we can use other more complicated algorithms that produce better assignments.

By adding by parts the above inequality with

$$T_{k_1} \le T_{k_2} \le \cdots \le T_{k_M}$$

we obtain condition (C1). By applying the above reasoning recursively, we have that condition (C1) is always satisfied. Note also that since containers in different rows are assigned to the shuttles condition (C2) is always satisfied.

Remark. As it can be seen from the proof, the order the shuttles arrive at the buffer is always the same during the application of Step 2; that is, shuttle i_1 arrives always first, shuttle i_2 second, etc., and the assignment always takes place when shuttle i_1 has finished loading the buffer. There is a danger that the gap between the first shuttle (shuttle i_1) and the last shuttle to become so large that the first shuttle approaches the last one with the possibility of collision. To avoid the danger of such a collision, we perform the following "trick": once the first shuttle is $T_b + \varepsilon$ seconds apart from the last shuttle we "assign" the second shuttle as the first one, the third as the second and so on, and perform the assignment every time the newly assigned first shuttle has finished loading the buffer. Here ε is a small positive design constant.

Similar to Proposition 1 we can prove the following:

Proposition 2: The dispatching algorithm DA guarantees that there are no delays due to factors (D1)-(D3) as long as there are at least *M* containers in different rows or, in the case where this is not true, there is at least one combination of *M* containers that satisfy (C2).

The proof of the above proposition is similar to the proof of Proposition 1. Since in the case where T_y is constant we have that condition (C2) is always satisfied and therefore,

Corollary 1: In the case where T_y is constant the dispatching algorithm DA guarantees that there are no delays due to factors (D1)-(D3) and therefore it maximizes throughput.

Comments:

1. In practice, the situation where no combination of shuttles/containers satisfies (C2) only happens when few containers are left to be loaded. Actually, the case where there is no combination of shuttles/containers that satisfy (C2) occurs with probability close to zero. Therefore, the proposed algorithm guarantees that there are no delays most of the time and it is optimal with large probability (close to 1). We therefore refer to this algorithm as near optimal. To test the above claim we run a simulation program as follows: we assigned randomly the containers to be loaded at different positions within the yard and we tested the above dispatching algorithm for different number *K* of containers. Only 0.02% of the different dispatching scenarios tested were found not to satisfy condition (C2).
2. Regarding Step 3 of the algorithm (i.e., the case where there are no *M* containers in different rows), it may happen that we examine all the different combinations of

shuttles/remaining containers until we find a solution that satisfies (C2). However, this happens only when there is small number of containers left to be loaded and thus the computations needed to examine condition (C2) for all possible combinations are practically feasible. In the same simulation program referred above, we found that the case where there are no M containers in different rows corresponds to the case where the containers left to be loaded are less than 1% of the total number of containers to be loaded.

3. Algorithm DA can be modified to deal with the cases where there are uncertainties or variations of the parameters of the model. Below we explain how.

   Robust Modification of DA: In the case where the parameters of the model are uncertain or vary with respect to their nominal values the following modifications:

   1. In Step 1, the shuttles should be initialized so they are T_b^u seconds apart, where T_b^u is a known upper bound for the parameter T_b, i.e., T_b^u satisfies $T_b \leq T_b^u$.
   2. In Steps 2 and 3 the inequalities

   (I.1)
   $$\begin{gathered} T_{i_1} < T_{i_2} < \cdots < T_{i_M} \\ T_{k_1} \leq T_{k_2} \leq \cdots \leq T_{k_M} \end{gathered}$$

   should be replaced by the following inequalities

   (I.2)
   $$\begin{gathered} T_{i_1}(\delta) < T_{i_2}(\delta) < \cdots < T_{i_M}(\delta), \forall \delta \in [0,1] \\ T_{k_1}(\delta) \leq T_{k_2}(\delta) \leq \cdots \leq T_{k_M}(\delta), \forall \delta \in [0,1] \end{gathered}$$

   where $T_i(\delta), T_k(\delta)$ are defined as follows

   $$\begin{gathered} T_i(0) = T_i^l, T_i(1) = T_i^u, T_i^l \leq T_i(\bar{\delta}) \leq T_i^u, \bar{\delta} \in (0,1), \\ T_k(0) = T_k^l, T_k(1) = T_k^u, T_k^l \leq T_k(\bar{\delta}) \leq T_k^u, \bar{\delta} \in (0,1) \end{gathered}$$

   Note that inequality (I.2) may not be satisfied for any pair of M containers. Thus if inequality (I.2) is not satisfied for the M containers that have been randomly in Steps 1 and 2 of the algorithm then we have to try again by randomly picking a different set of containers.

   Finally, note that

   $$T_k^l = \frac{d_{1k}}{v_{unloaded}^u} + F^l(Y_k, C_k) + \frac{d_{2k}}{v_{loaded}^u}, T_k^u = \frac{d_{1k}}{v_{unloaded}^l} + F^u(Y_k, C_k) + \frac{d_{2k}}{v_{loaded}^l}$$

   and

   $$T_i^u = T_b^u + \frac{d_i}{v_{loaded}^l}, T_i^l = T_b^l + \frac{d_i}{v_{loaded}^u}$$

   In the above equations, the superscripts u, l stand for the upper and lower bound of the corresponding variable.

4. The proposed algorithm imposes implicitly the following two assumptions:

   (A1) The configuration and dimensions of the GR unit, the number M of shuttles and processing times T_b and its upper bound T_b^u must be such that it is possible to initialize the shuttles' positions so that they are T_b or T_b^u seconds apart.

   (A2) The number of shuttles M is less or equal to the number of rows in the unit.

Assumption (A1) excludes non-realistic situations such as the situation where T_b is orders of magnitude larger than the traveling times T_k, in which case it is not possible to initialize the shuttles' positions so that they are T_b seconds apart.

6. Simulations of a GR System

In order to evaluate the performance of the GR system and the algorithms proposed we performed simulations using a typical container yard configuration. The size of the yard was chosen to be 400 by 180 feet, which is equivalent of 20x36=720 storing spots for 20 feet long, 5 feet wide containers. Container yards of the aforementioned sizes are typically met in container terminals and they are served by two-three quay cranes. Four different simulation sets were performed. In all four simulations, the container positions were chosen randomly; up to five high container storage was assumed. The simulation parameters are shown in Table 1. All parameters were Gaussian random variables with average the nominal value of the parameter and variance equal to 10% of the nominal value. The four simulation tests are described below:

Simulation 1 (Random Assignment/No Reshuffling): In this simulation, the containers were randomly assigned to empty shuttles. No reshuffling was assumed, i.e., there were no containers stored above the containers to be loaded.

Simulation 2 (DA/No Reshuffling): In this simulation, the containers were assigned using the dispatching algorithm DA. No reshuffling was assumed, i.e., there were no containers stored above the containers to be loaded.

Simulation 3 (Random Assignment/With Reshuffling): In this simulation, the containers were randomly assigned to empty shuttles. Reshuffling was simulated in this simulation. More precisely, one third of the containers to be loaded had one container stored above them, one third had two containers stored above and one third had no container stored above.

Simulation 4 (DA/With Reshuffling): In this simulation, the containers were assigned using the dispatching algorithm DA. Reshuffling was simulated in this simulation. More precisely, one third of the containers to be loaded had one container stored above them, one third had two containers stored above and one third had no container stored above.

Table 1: Parameter values used in the simulations

Parameter	Description	Value	Units	Variance
T_b	quay buffer loading/unloading time	30 (120 moves per hour)	seconds	10%
$v_{unloaded}$	shuttle speed (unloaded)	10 (6.8mph)	ft/sec	10%
v_{loaded}	shuttle speed (loaded)	5 (3.4mph)	ft/sec	10%
$\bar{v}_{unloaded}$	shuttle hoist speed (unloaded)	10 (6.8mph)	ft/sec	10%
$\bar{v}_{loaded}$	shuttle hoist speed (loaded)	5 (3.4mph)	ft/sec	10%
$\bar{T}_c$	time needed to attach the container to the shuttle		seconds	10%

Figures 3, 5, 7, and 9, plot the GR unit throughput (measured in number of containers loaded in the quay buffer per hour) versus the GR unit size for different number of shuttles, for the cases of simulation 1,2,3, and 4, respectively. Figures 4, 6, 8, and 10, plot the total GR system throughput (measured in total number of containers loaded in the quay buffers per hour) versus the number of GR unit for different number of shuttles, for the cases of simulation 1,2,3, and 4, respectively. Analyzing Figures 3-10, we can conclude that:

- The productivity of the GR system heavily depends on the number of shuttles used. The relation between the number of GR units and total throughput is linear in most cases and always strictly increasing. GR systems with 12 units can achieve more than 1000 containers per hour while GR systems with 3 units cannot exceed 300 containers per hour. The price paid for the high productivity rates achieved when a large number of units is used is that the operational cost increases on the one hand (notice that a system with 12 units and 5 shuttles per unit needs a total of 12x5=60 shuttles, while a system with 3 units with 10 shuttles per unit needs a total of 3x10=30 shuttles); on the other hand using large number of units may complicate the dispatching algorithm used for the vehicles that transfer the containers from/to the quay buffers to/from the quay cranes.
- The proposed DA algorithm improves considerably the productivity, especially when reshuffling is needed. Notice though that when there is no reshuffling and the number of shuttles is large, the productivity of the proposed algorithm is almost identical to the one achieved by using random assignment. This is since, when there is no reshuffling, there are no delays due to factor (D3), while the delays due to factors (D1)-(D2) are negligible because there is redundancy of shuttles.
- Larger GR units can achieve similar productivity rates as smaller ones at the expense of increasing the number of shuttles used.

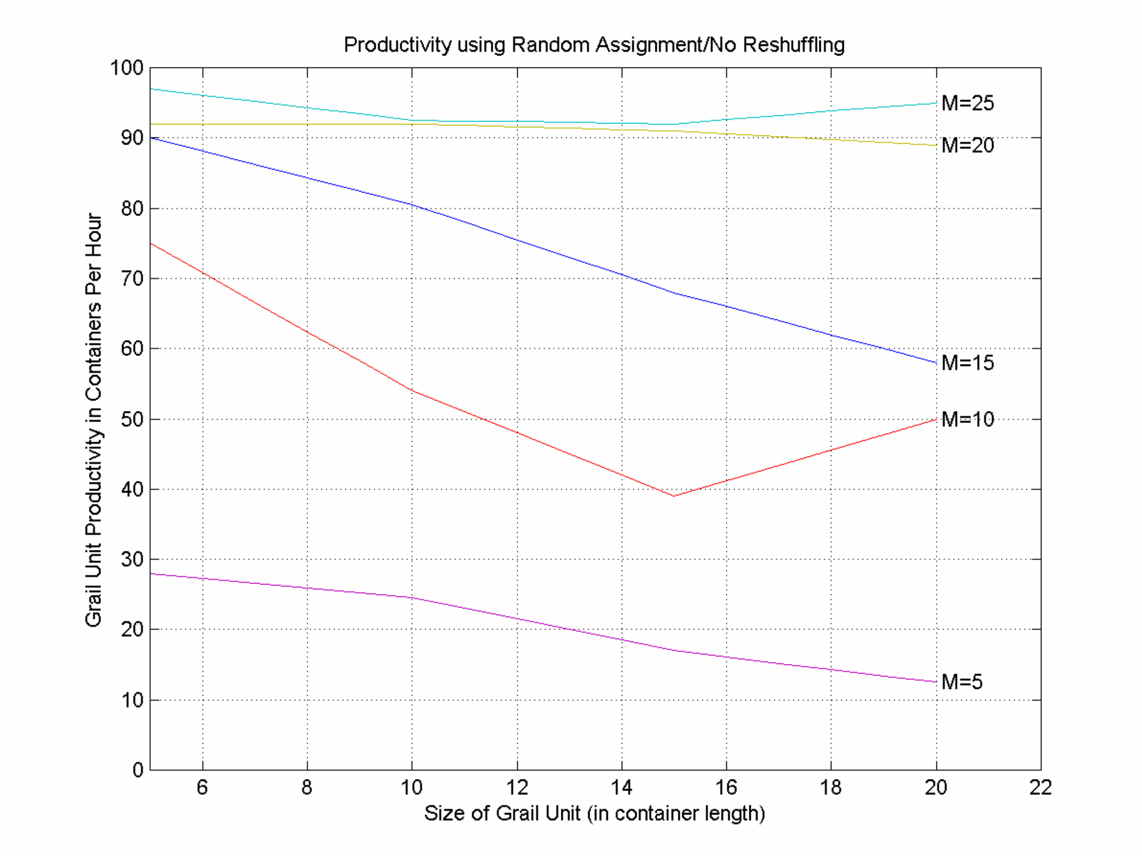

Figure 3:Productivity of GR units for different GR unit sizes and different number M of shuttles – Random Assignment/No Reshuffling

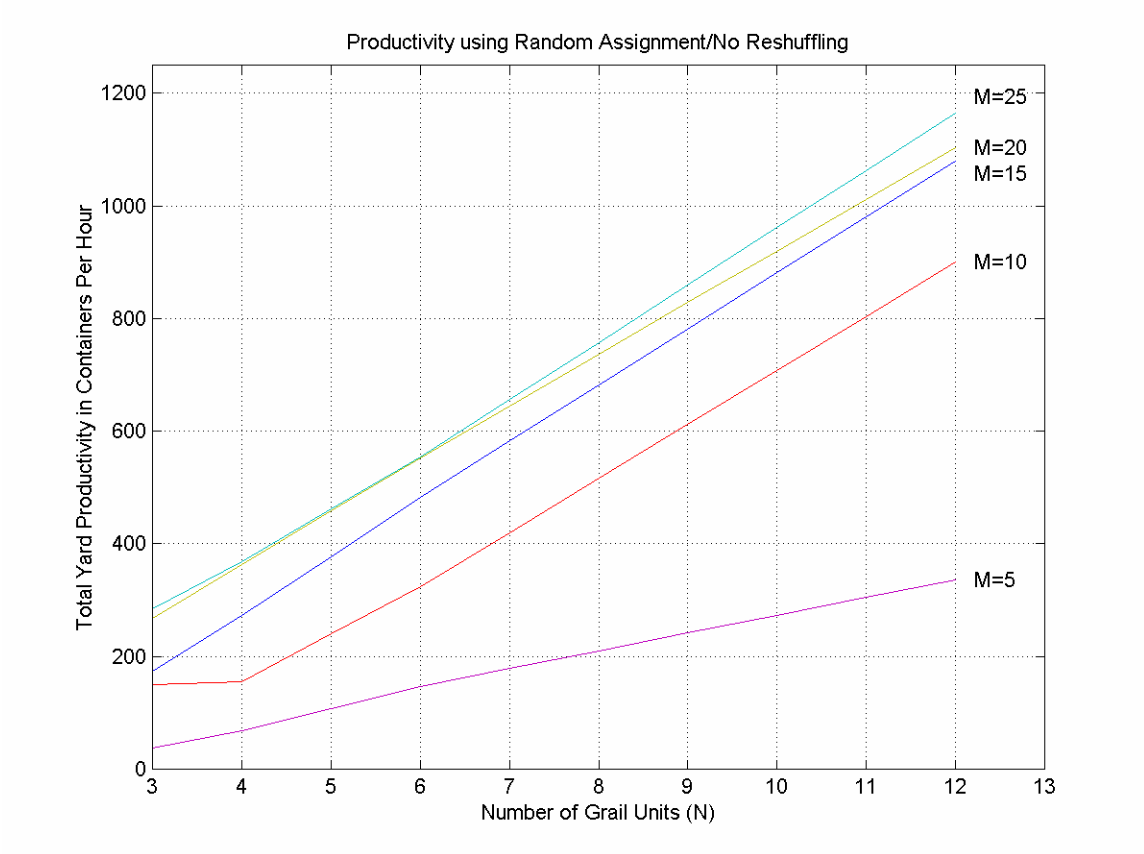

Figure 4:Productivity of GR system for different number N of GR units and different number M of shuttles per unit– Random Assignment/No Reshuffling

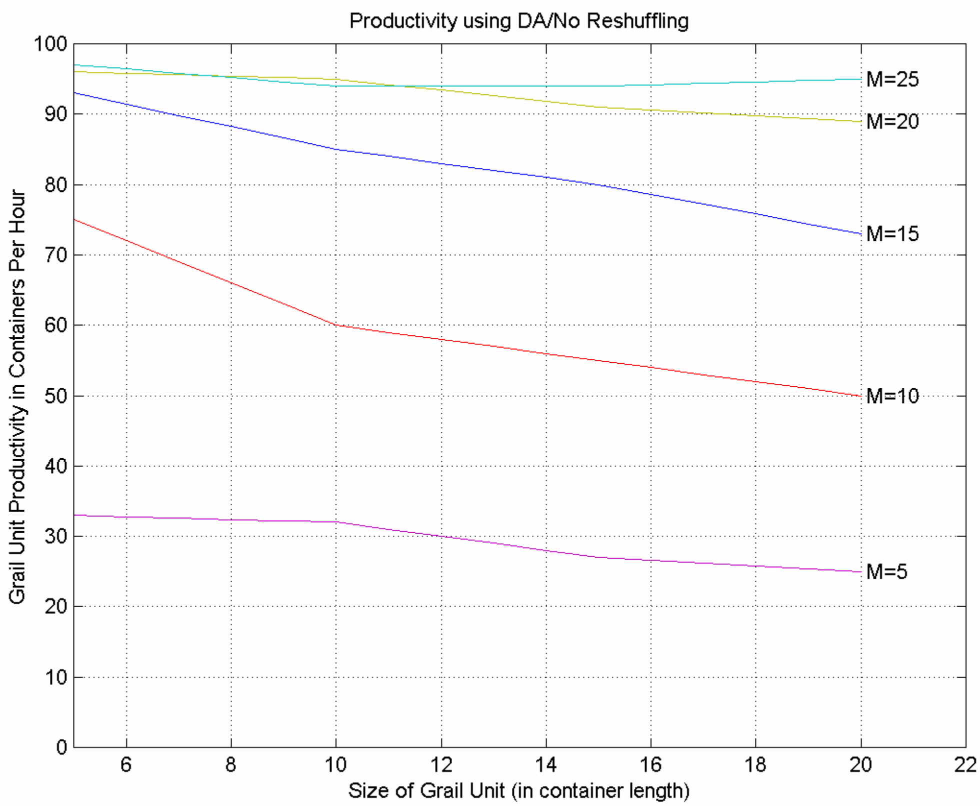

Figure 5:Productivity of GR units for different GR unit sizes and different number M of shuttles – DA/No Reshuffling

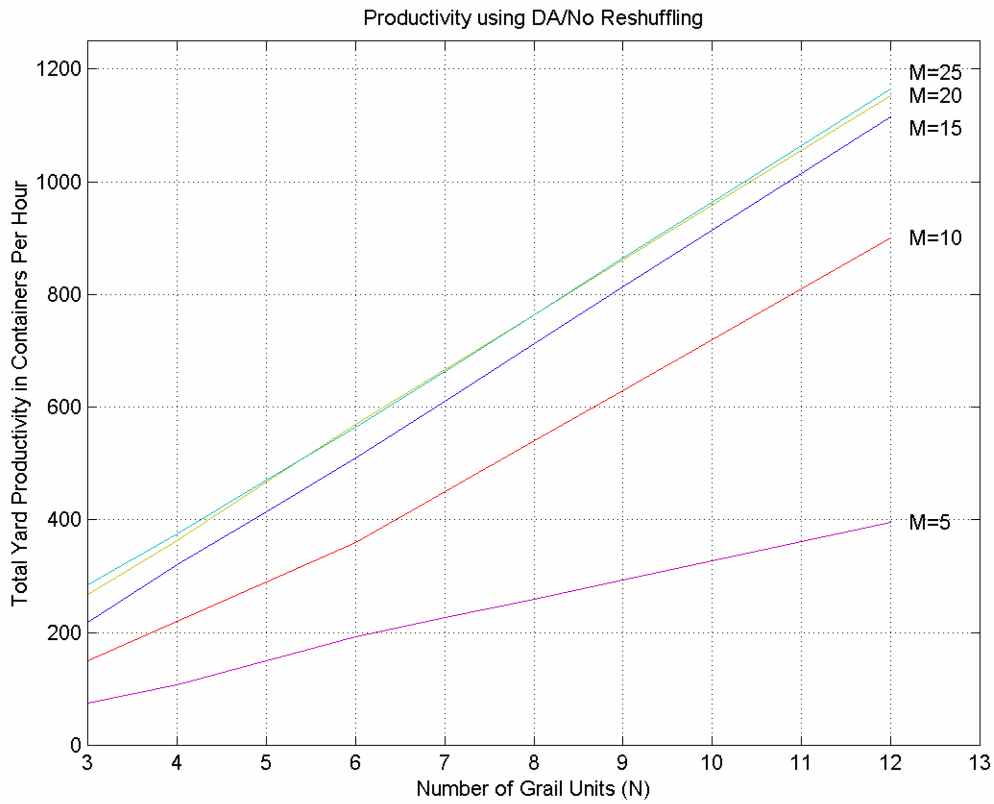

Figure 6:Productivity of GR system for different number N of GR units and different number M of shuttles per unit– DA/No Reshuffling

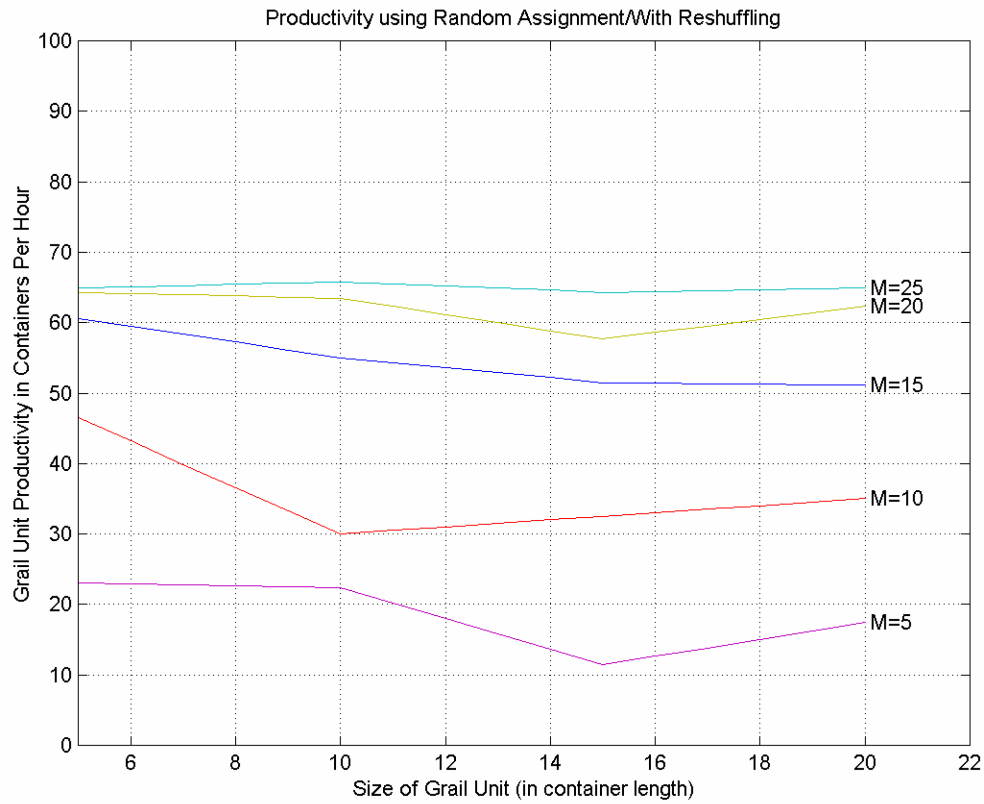

Figure 7:Productivity of GR units for different GR unit sizes and different number M of shuttles – Random Assignment/With Reshuffling

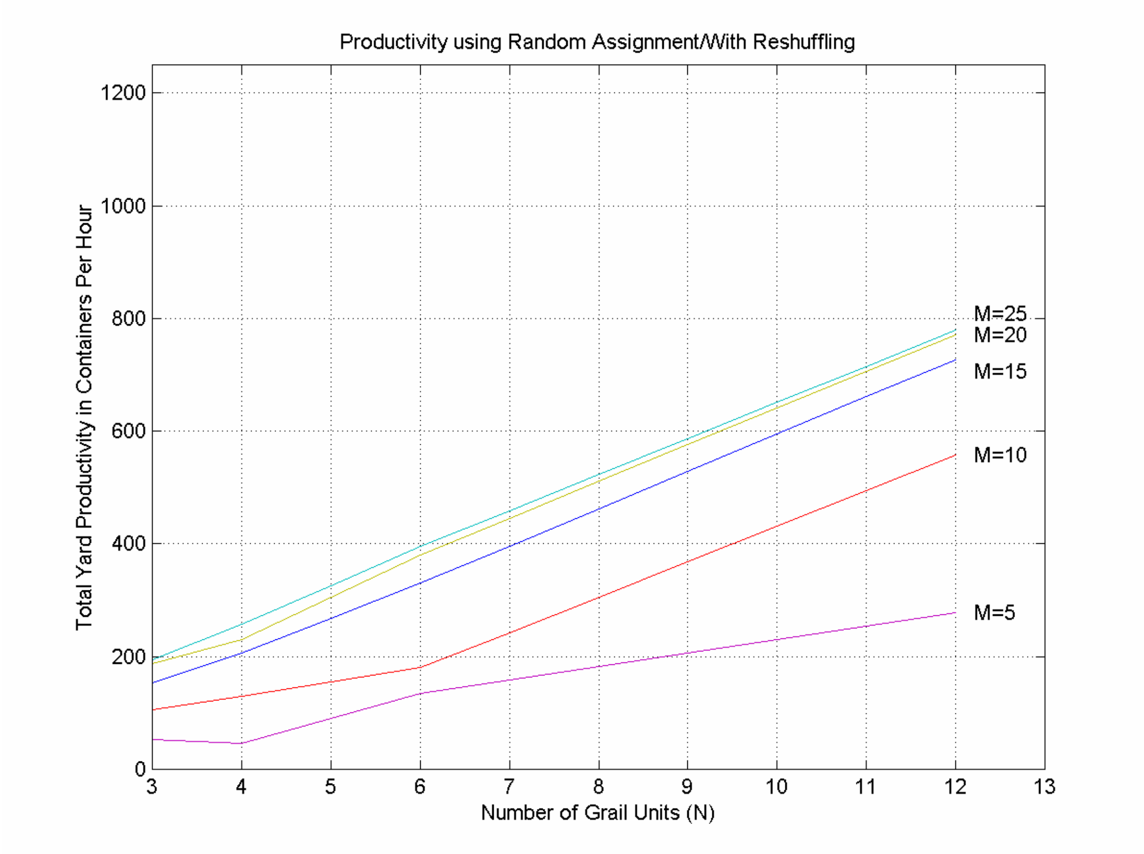

Figure 8:Productivity of GR system for different number N of GR units and different number M of shuttles per unit– Random Assignment/With Reshuffling

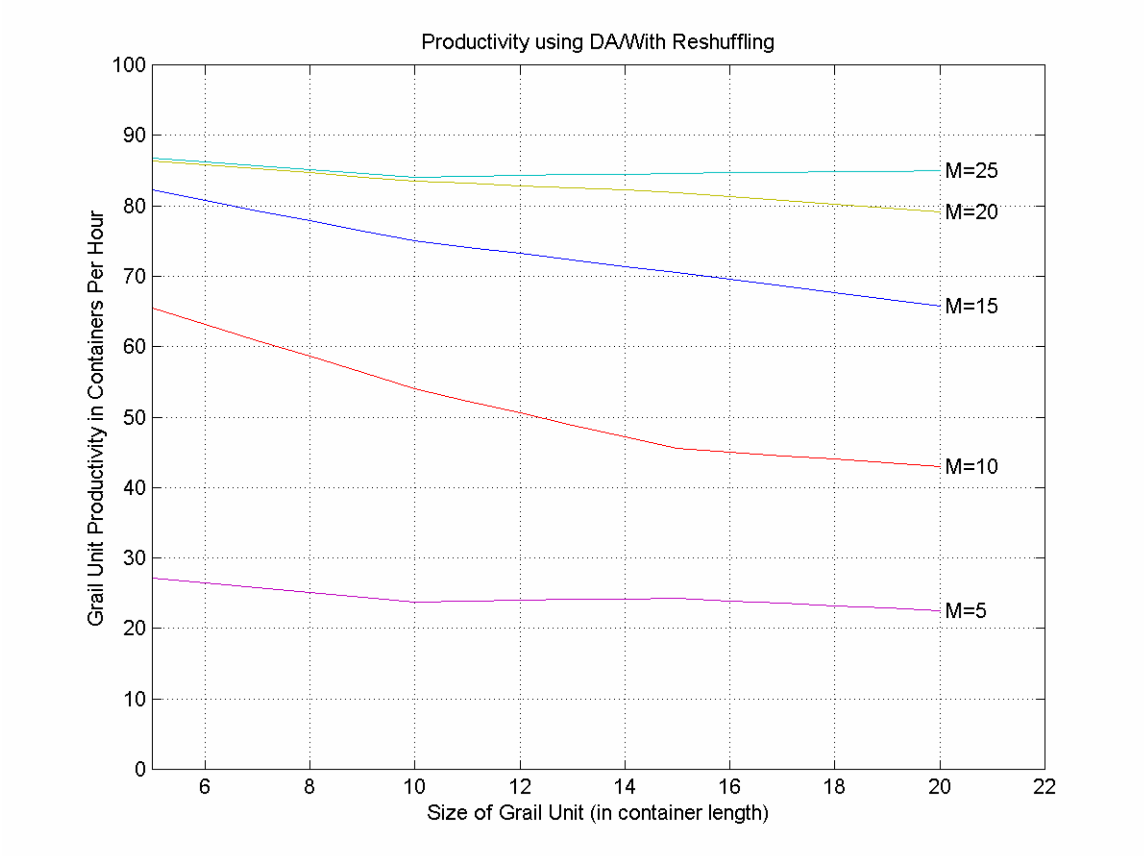

Figure 9:Productivity of GR units for different GR unit sizes and different number M of shuttles – DA/With Reshuffling

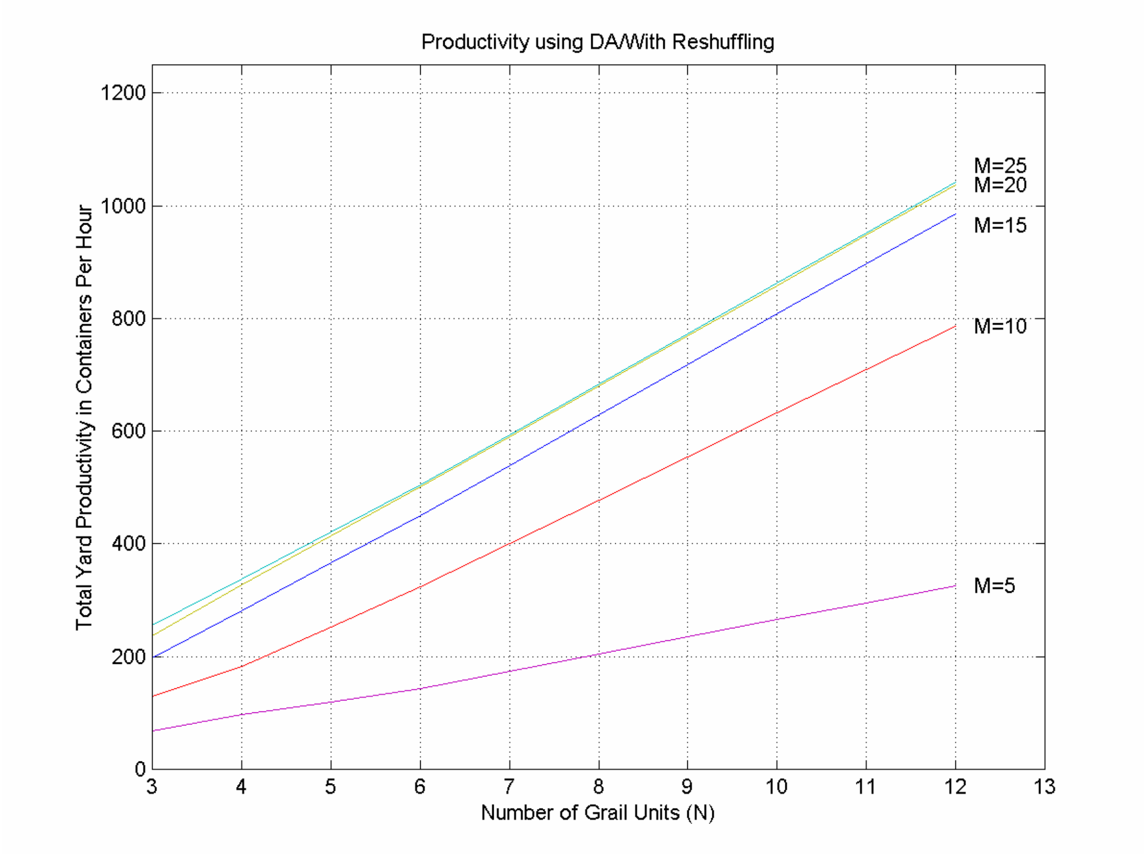

Figure 10:Productivity of GR system for different number N of GR units and different number M of shuttles per unit– DA/With Reshuffling

7. Evaluation of GR Systems Operating in Automated Container Yards

One of the crucial problems in today's maritime transportation is the development of appropriate cargo handling technologies that will make possible the loading/unloading of a ship in less than 24 hours. In this section, we evaluate the capabilities of GR systems when they are operating in an automated container yard to meet the aforementioned goal of loading/unloading a ship in less than 24 hours.

For this purpose, we propose and evaluate a typical conceptual automated container yard, shown in Figure 11. The yard consists of the gate where containers are transferred to/from the yard using trucks, the railway for transferring containers using rail, a GR system consisting of N units, the berth area where cranes load/unload containers to/from the ship, and the truck/AGV interface where containers are temporarily placed in order to be transferred to the gate or the yard. Trucks are used to transfer the containers to/from the gate and AGVs to transfer the containers to/from the yard.

In our study, we assume that there are 8 GR units (N=8) that communicate with other parts of the yard through buffers: 1a, 1b, 2a, 2b, …, 8b. There are transit roads every between every two units. These transit roads are used for transferring containers – using AGVs – to/from the truck/AGV interface directly to the ship. It is assumed that 30% of containers is directly delivered from the buffer to the ship by AGVs. The cargo that has to be stored in the yard goes through GR units. The first 4 units are used for transferring containers between the truck/AGV interface and the GR system and the rest 4 units are used for transferring containers between the GR system and the ship (quay cranes). In other words, the units 1, 2, 5, 6 transfer and store export containers and the units 3, 4, 7, 8 transfer and store import containers. One AGV in one cycle goes from the truck/AGVs buffer interface with an export container, leaves the container at the GR buffer (either 1a or 2a) and travels empty to the GR buffer (either 3a or 4a) where it is loaded by an import container and travels back loaded to the truck/AGVs buffer interface.

Thus, the cargo stored in the first two GR units is not bound for the ship that is currently loaded/unloaded. Similarly, one AGV in one cycle goes from the GR buffer (either 5b or 6b) with an export container, leaves the container at the quay crane, picks an import container from the quay crane and travels to the GR buffer (either 7b or 8b) where it loads the container to the GR buffer and travels empty back to the GR buffers 5b or 6b. In the case where there are no import containers available at the quay crane the AGV returns empty to buffers 5b or 6b where it is loaded with a new export container. Similarly, in the case where there are no export containers at the GR buffers 5b and 6b the AGV transfers containers from the quay crane to buffers 7b and 8b.

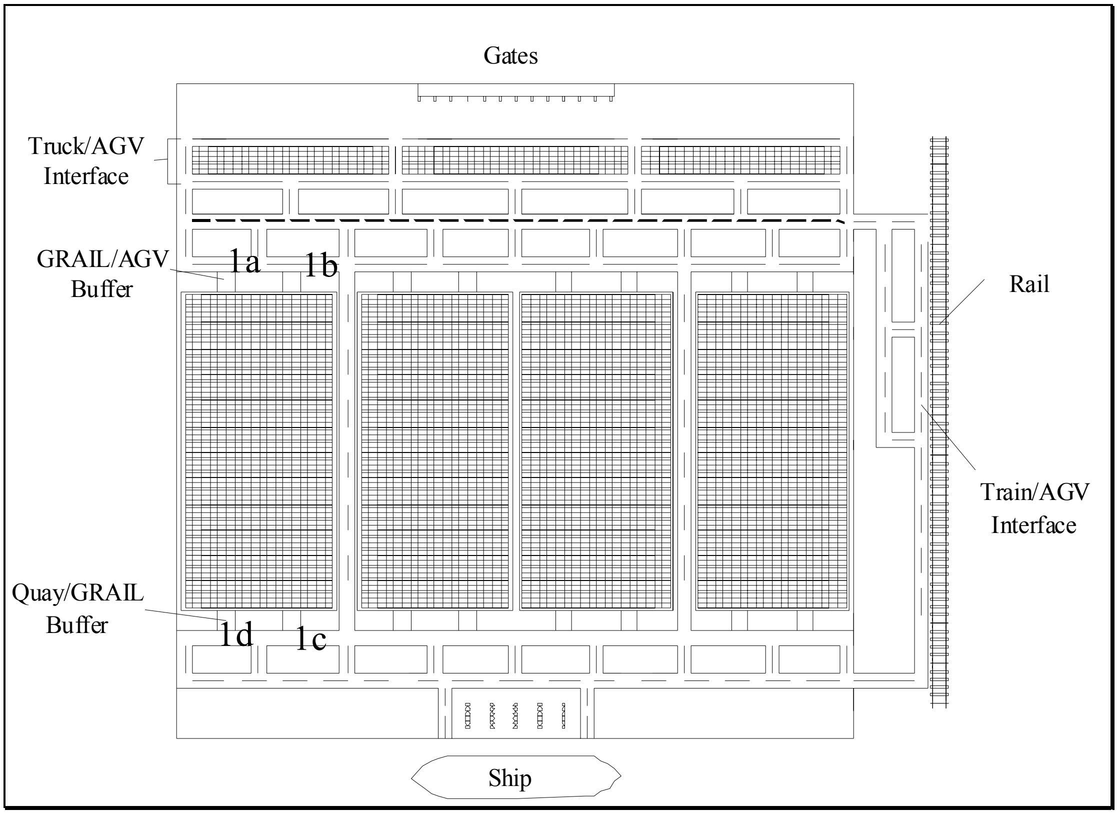

Figure 11: The conceptual automated container yard assumed in the simulations

We assume the following:

1. The ships have all the same carrying capacity of 8,000 TEUs and they have to be 85% loaded. That is, it is assumed that the ships are scheduled to carry 6,800 TEUs.
2. The export containers start arriving 2 days before the ship is scheduled to arrive, with arrival rates 0.2, 0.5, 0.3, that is 20% of the containers arrive two days in advance, 50% of the containers arrive one day in advance and 30% of the containers arrive the day that the ship arrives. We also assume that there three liners that call at the terminal on a regular basis (period is three days). Thus, every day 6,800 TEUs are arriving at the yard, 20% of which correspond to the ship that will arrive in two days, 50% correspond to the ship that will arrive the next day and 30% correspond to the ship that has already arrived. We also assume that 4,080 TEUs arrive by trucks and the rest 2720 TEUs arrive by train. 70% of the containers are placed in the yard (30% is directly delivered to the ship by AGVs).

Our assumptions regarding the arrival rates of the containers are summarized in Tables 2 and 3.

Table 2: The cumulative numbers of export containers, which arrive by trucks and trains

	Every day
The number of containers that arrive by trucks	4080 TEUs: 30% of this amount is directly delivered to the vessel; 50% of cargo arrives one day in advance of the bound vessel, 20% of cargo arrives two days in advance of the bound vessel
The number of containers that arrive by trains	2720 TEUs: 30% of this amount is directly delivered to the vessel; 50% of cargo arrives one day in advance of the bound vessel, 20% of cargo arrives two days in advance of the bound vessel

Table 3: The cumulative numbers of import containers that are retrieved by trucks and trains

	Every day
The number of containers that are retrieved by trucks	4080 TEUs: 50% of this amount is directly retrieved from the vessel; 30% of cargo is retrieved from the storage (it came one day ago), 20% of cargo is retrieved from the storage (it came two days ago)
The number of containers that are retrieved by trains	2720 TEUs: 50% of this amount is directly retrieved from the vessel; 30% of cargo is retrieved from the storage (it came one day ago), 20% of cargo is retrieved from the storage (it came two days ago)

We[7] also assume that at the buffers containers are served by the same rate (incoming and outgoing rates are the same) – the number of AGVs and yard cranes can be determined to avoid bottlenecks. Thus the incoming export container arrival rate to the GR/AGV buffers (denoted as 1a and 1b) can be calculated as the sum of the incoming trucks rate (0.7*2040/24*60 = 0.99/min) + the incoming containers rate by train (0.7*1360/24*60=0.66/min). The above number is equal to 1.65 containers/min for the whole GR system or 99/2=49.5 containers/hour per buffer.

At the same time, the other GR unit has to receive import containers unloaded from the ship and transferred by AGVs to the Quay/GR buffers. It is assumed that 50% of import containers are transferred directly to trucks/AGV buffers and to the rail. However, due to imbalance we will assume that 30% will be transferred by AGVs directly from the ship (during ship service) and 70% (2380 FEUs) goes through storage until the ship is served and gone and then it is transferred to trucks/AGV buffers and to the rail from GR unit by AGVs. These containers are unloaded from the ship with the following rate: There are 5 ship cranes with the speed 50 moves/hr = 0.83 moves/hr. The average crane service rate for combined loading/unloading operation is 0.70 container/min and therefore the total service rate is 5*0.70=3.5 containers/min. With this rate, ship is served for 16 hours.

[7] We assume that 40 feet containers are only used, thus each container corresponds to 2 TEU's.

It is assumed that 70% of these containers are placed in the storage; thus, the incoming rate at the quay buffers is 0.7*3.5=2.45 containers/min or 73.5 containers/hour/quay buffer.

Thus the question in hand is whether the GR units can handle 49.5 containers per hour at the gate buffer when the unit and 73.5 containers per hour at the quay buffer. Since both loading and unloading is considered four different cases have to be evaluated:

1. The case where export containers are coming through the gate buffer and are stored in the GR unit.
2. The case where the GR unit is unloaded and the containers are loaded to the gate buffer.
3. The case where import containers are coming through the quay buffer and are stored in the GR unit.
4. The case where the GR unit is unloaded and the containers are loaded to the quay buffer.

Since the GR unit is symmetrical cases 1 and 3 and cases 2 and 4 are identical. Therefore only two different cases will be considered: the case where the containers are picked from the buffer and are stored in the unit (loading) and the case where the containers are picked from the unit and are placed in the buffer (unloading).

A simulation program has been developed for evaluating the GR unit capabilities for different number of shuttles. Each GR unit has capacity equal to 2592 TEUs (this corresponds to a total of 2592*4 = 10368 TEUs for the import (first four units) and the export (last four units) storage areas or 10368*2=20736 TEUs for the whole area covered by the GR system. Assuming that up to 4 containers-high stacks are allowed, the total size of each unit is assumed to be 9 containers wide and 36 containers long.

Since every day containers that correspond to three different ships the following policy was assumed in the simulations: each container being handled is assigned randomly to one of the three ships with probability 0.2, 0.5 and 0.3, respectively. The containers are also assumed stored randomly within the GR unit; however, each stack of containers corresponds to containers that are to be loaded on the same ship. No reshuffling is assumed for the containers (that is, the particular order at which the containers are loaded in the ship is assumed to be of no significance).

In Figure 12, we plot the unit productivity for different number of shuttles for both the cases of loading and unloading. The simulation parameters used are the same as in Table 1 of the previous chapter. In Figure 13, the average idle time is plotted. By average idle time, we define the average of the percentage of the time lost from delays due to factors (D1)-(D3) per shuttle.

Analyzing Figures 12 and 13 we can see that:

1. GR units with more than 9 shuttles for the case where the containers are transferred from/to the truck/AGV area to/from the GR system and more than 15 shuttles for the case where the containers are transferred from/to the quay cranes

to/from the GR system can achieve the goal of loading/unloading the ship in less than 24 hours.

2. The average idle time, i.e., the time wasted in delays is negligible. However, this time increases as the number of shuttles increases. This is mainly due to the usage of the dispatching algorithm proposed in this report.
3. The GR system under study is capable of achieving better than 24 hours loading/unloading by increasing the number of shuttles used. This is possible of course provided that faster quay cranes are used and the arrival rates of the import containers are larger than the ones assumed in our simulations.

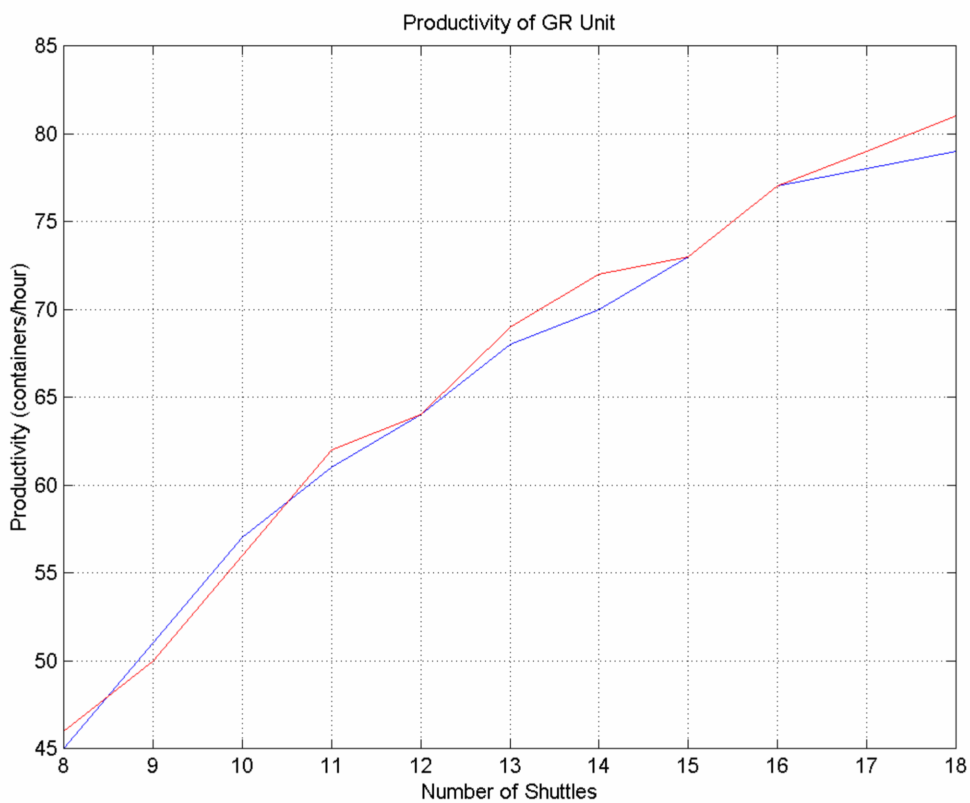

Figure 12: Productivity of GR Unit for different number of shuttles (loading: red curve, unloading: blue curve).

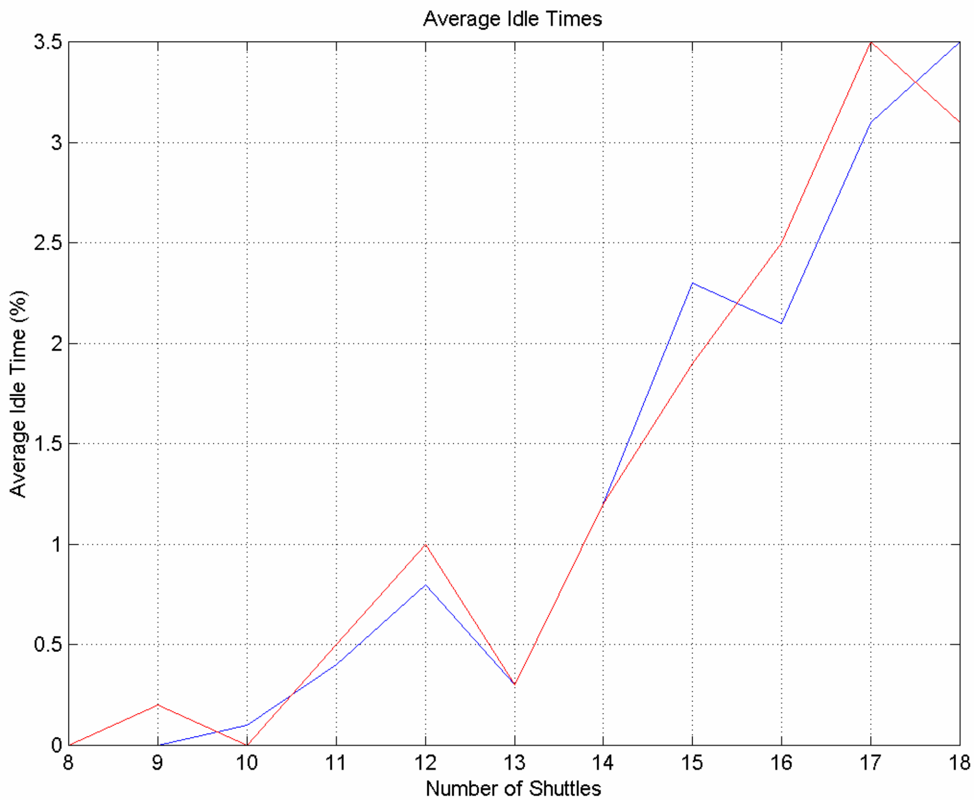

Figure 13: Average Idle Time of GR Unit for different number of shuttles (loading: red curve, unloading: blue curve).

8. Conclusions and Recommendations

A container terminal using Grid Rail (GR) units or modules was developed based on the GRAIL system designed by Sea-Land/August Design Inc. a non-automated version of which is operating in the Port of Hong Kong. The GR units offer the advantages of high storage density, fast loading/unloading, flexibility and reliability and no interference between manual and automated operations. Moreover, contrary to other automated container concepts, the simplicity of GR operations makes it possible to develop optimal or near optimal dispatching algorithms.

The focus of this study was to design the operations within the GR units so that a minimum number of shuttles are used to serve the ship and gate buffers. A dispatching algorithm is developed that synchronizes the motion of the shuttles in order to serve the buffers of the GR units, minimize delays and maximize throughput. The GR units are used as part of an automated container terminal. Simulations demonstrate that the GR unit concept can be used in container terminals to improve productivity and reduce cost while utilizing much less land than conventional container terminals.

References

[1] Aarts E., Korst J. (1989), *Simulated Annealing and Boltzmann Machines,* John Wiley & Sons.

[2] Al-Sultan K.S., and Bozer Y.A. (1998), “Network Configuration and Machine Layout in Fixed-Path Material Handling Systems,” *Annals of Operation Research,* Vol.76.

[3] Co C.G., Tanchoco J.M.A., “A Review of Research on the AGVS Vehicle Management”, *Engineering Costs and Production & Econom*ics, Vol.21, 1991, PP 3542.

[4] Egbelu P.J., Tanchoco J.M.A., "Characterization of automatic guided vehicle dispatching rules,” International Journal of production Research, Vol. 22, No. 3, 1984, PP.359-374.

[5] Ioannou P.A., et al., (1999), *Cargo Handling Technologies,* Technical Report, University of Southern California.

[6] Jula H., et al., (2000), “Container Terminals Using Automated Shuttles Driven by Linear Motors,” Accepted to be appeared in IFAC Symposium on Control in Transportation Systems 2000, Braunschweig, Germany, June 2000.

[7] Jula H., et al., (2000), “Containers Sequencing for AGVs in an Automated Container Terminal”, paper under preparation

[8] Peters B. A., Smith J. S., and Venkatesh S. (1996), “A Control Classification of Automated Guided Vehicle Systems,” *International Journal of Industrial Engineering,* Vol.3, No.1, pp. 29-39.

[9] Taleb-Ibrahimi M. (1989), *Modeling and Analysis of Container Handling in Ports,* Ph.D. Thesis, Department of Industrial Engineering and Operation Research, University of California, Berkeley, USA 1989.

[10] URL: http://www.tekem.co.il/English/agvpage.htm

[11] URL: http://www.august-design.com

[12] URL: http://ahs.volpe.dot.gov/avcsdoc

[13] URL: http://www.agve.se/

[14] URL: http://www.noell.de/